

Clustering on Multi-Layer Graphs via Subspace Analysis on Grassmann Manifolds

Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst and Nikolai Nefedov

Abstract—Relationships between entities in datasets are often of multiple nature, like geographical distance, social relationships, or common interests among people in a social network, for example. This information can naturally be modeled by a set of weighted and undirected graphs that form a global multi-layer graph, where the common vertex set represents the entities and the edges on different layers capture the similarities of the entities in term of the different modalities. In this paper, we address the problem of analyzing multi-layer graphs and propose methods for clustering the vertices by efficiently merging the information provided by the multiple modalities. To this end, we propose to combine the characteristics of individual graph layers using tools from subspace analysis on a Grassmann manifold. The resulting combination can then be viewed as a low dimensional representation of the original data which preserves the most important information from diverse relationships between entities. We use this information in new clustering methods and test our algorithm on several synthetic and real world datasets where we demonstrate superior or competitive performances compared to baseline and state-of-the-art techniques. Our generic framework further extends to numerous analysis and learning problems that involve different types of information on graphs.

Index Terms—Multi-layer graphs, subspace representation, Grassmann manifold, clustering.

I. INTRODUCTION

GRAPHS are powerful mathematical tools for modeling pairwise relationships among sets of entities; they can be used for various analysis tasks such as classification or clustering. Traditionally, a graph captures a single form of relationships between entities and data are analyzed in light of this one-layer graph. However, numerous emerging applications rely on different forms of information to characterize relationships between entities. Diverse examples include human interactions in a social network or similarities between images or videos in multimedia applications. The multimodal nature of the relationships can naturally be represented by a set of weighted and undirected graphs that share a common set of vertices but with different edge weights depending on the type of information in each graph. This can then be represented by a multi-layer or multi-view graph which gathers all sources of information in a unique representation. Assuming that all the graph layers are informative, they are likely to provide complementary information and thus to offer richer

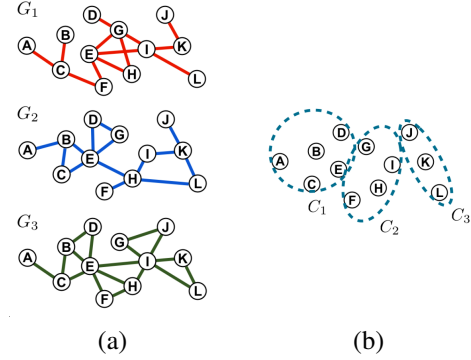


Fig. 1. (a) An illustration for a three-layer graph G , whose three layers $\{G_i\}_{i=1}^3$ share the same set of vertices but with different edges. (b) A potential unified clustering $\{C_k\}_{k=1}^3$ of the vertices based on the information provided by the three layers.

information than any single layer taken in isolation. We thus expect that a proper combination of the information contained in the different layers leads to an improved understanding of the structure of the data and the relationships between entities in the dataset.

In this paper, we consider a M -layer graph G with individual graph layers $G_i = \{V, E_i, \omega_i\}$, $i = 1, \dots, M$, where V represents the common vertex set and E_i represents the edge set in the i -th individual graph G_i with associated edge weights ω_i . An example of a three-layer graph is shown in Fig. 1 (a), where the three graph layers share the same set of 12 vertices but with different edges (we assume unit edge weights for the sake of simplicity). Clearly, different graph layers capture different types of relationships between the vertices, and our objective is to find a method that properly combines the information in these different layers. We first adopt a subspace representation for the information provided by the individual graph layers, which is inspired by the spectral clustering algorithms [1], [2], [3]. We then propose a novel method for combining the multiple subspace representations into one representative subspace. Specifically, we model each graph layer as a subspace on a Grassmann manifold. The problem of combining multiple graph layers is then transformed into the problem of efficiently merging different subspaces on a Grassmann manifold. To this end, we study the distances between the subspaces and develop a new framework to merge the subspaces where the overall distance between the representative subspace and the individual subspaces is minimized. We further show that our framework is well justified by results from statistical learning theory [4], [5]. The proposed method is a dimensionality reduction algorithm for the original data;

X. Dong, P. Frossard and P. Vandergheynst are with Signal Processing Laboratories (LTS4/LTS2), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (e-mail: xiaowen.dong@epfl.ch; pascal.frossard@epfl.ch; pierre.vandergheynst@epfl.ch).

N. Nefedov is with Signal and Information Processing Laboratory, Eidgenössische Technische Hochschule Zürich (ETH Zürich), Zurich, Switzerland (e-mail: nefedov@isi.ee.ethz.ch).

it leads to a summarization of the information contained in the multiple graph layers, which reveals the intrinsic relationships between the vertices in the multi-layer graph.

Various learning problems can then be solved using these relationships, such as classification or clustering. Specifically, we focus in this paper on the clustering problem: we want to find a unified clustering of the vertices (as illustrated in Fig. 1 (b)) by utilizing the representative subspace, such that it is better than clustering achieved on any of the graph layers G_i independently. To address this problem, we first apply our generic framework of subspace analysis on the Grassmann manifold to compute a meaningful summarization (as a representative subspace) of information contained in the individual graph layers. We then implement a spectral clustering algorithm based on the representative subspace. Experiments on synthetic and real world datasets demonstrate the advantages of our approach compared to baseline algorithms, like the summation of individual graphs [6], as well as state-of-the-art techniques, such as co-regularization [7]. Finally, we believe that our framework is beneficial not only to clustering, but also to many other data processing tasks based on multi-layer graphs or multi-view data in general.

This paper is organized as follows. We first review the related work and summarize the contribution of the paper in Section II. In Section III, we describe the subspace representation inspired by spectral clustering, which captures the characteristics of a single graph. In Section IV, we review the main ingredients of Grassmann manifold theory, and propose a new framework for combining information from multiple graph layers. We then propose our novel algorithm for clustering on multi-layer graphs in Section V, and compare its performance with other clustering methods on multiple graphs in Section VI. Finally, we conclude in Section VII.

II. RELATED WORK

In this section we review the related work in the literature. First, we describe briefly graph-based clustering algorithms, with a particular focus on the methods that have subspace interpretations. Second, we summarize the previous works built upon subspace analysis and the Grassmann manifold theory. Finally, we report the recent progresses in the field of analysis of multi-layer graphs or multi-view data.

Clustering on graphs has been studied extensively due to its numerous applications in different domains. The works in [8], [9] have given comprehensive overviews of the advancements in this field over the last few decades. The algorithms that are based on spectral techniques on graphs are of particular interest, typical examples being spectral clustering [1], [2], [3] and modularity maximization via spectral method [10], [11]. Specifically, these approaches propose to embed the vertices of the original graph into a low dimensional space, usually called the spectral embedding, which consists of the top eigenvectors of a special matrix (graph Laplacian matrix for spectral clustering and modularity matrix for modularity maximization). Due to the special properties of these matrices, clustering in such low dimensional spaces usually becomes trivial. Therefore, the corresponding clustering approaches can be interpreted

as transforming the information on the original graph into a meaningful subspace representation. Another example is the Principal Component Analysis (PCA) interpretation on graphs described in [12], which links the graph structure to a subspace spanned by the top eigenvectors of the graph Laplacian matrix. These works have inspired us to consider the subspace representation in Section III.

In the past few decades, subspace-based methods have been widely used in classification and clustering problems, most notably in image processing and computer vision. In [13], [14], the authors have discovered that human faces can be characterized by low-dimensional subspaces. In [15], the authors have proposed to use the so-called “eigenfaces” for recognition. Inspired by these works, researchers have been particularly interested in data where data points of the same pattern can be represented by a subspace. Due to the growing interests in this field, there is an increasingly large number of works that use tools from the Grassmann manifold theory, which provides a natural tool for subspace analysis. In [16], the authors have given a detailed overview of the basics of the Grassmann manifold theory, and developed new optimization techniques on the Grassmann manifold. In [17], the author has presented statistical analysis on the Grassmann manifold. Both works study the distances on the Grassmann manifold. In [18], [4], the authors have proposed learning frameworks based on distance analysis and positive semidefinite kernels defined on the Grassmann manifold. Other recent representative works include the studies in [19], [20] where the authors have proposed to find optimal subspace representation via optimization on the Grassmann manifold, and the analysis in [21] where the authors have presented statistical methods on the Stiefel and Grassmann manifolds for applications in vision. Similarly, the work in [22] has proposed a novel discriminant analysis framework based on graph embedding for set matching, and the authors in [23] have presented a subspace indexing model on the Grassmann manifold for classification. However, none of the above works considers datasets represented by multi-layer graphs.

At the same time, multi-view data have attracted a large amount of interest in the learning research communities. These data form multi-layer graph representations (or multi-view representations), which generally refer to data that can be analyzed from different viewpoints. In this setting, the key challenge is to combine efficiently the information from multiple graphs (or multiple views) for learning purposes. The existing techniques can be roughly grouped into the following categories. First, the most straightforward way is to form a convex combination of the information from the individual graphs. For example, in [24], the authors have developed a method to learn an optimal convex combination of Laplacian kernels from different graphs. In [25], the authors have proposed a Markov mixture model, which corresponds to a convex combination of the normalized adjacency matrices of the individual graphs, for supervised and unsupervised learning. In [26], the authors have presented several averaging techniques for combining information from the individual graphs for clustering. Second, following the intuitive approaches in the first category, many existing works aim at finding a unified

representation of the multiple graphs (or multiple views), but using more sophisticated methods. For instances, the authors in [6], [27], [28], [29], [30], [31] have developed several joint matrix factorization approaches to combine different views of data through a unified optimization framework, where the authors in [32] have proposed to find a unified spectral embedding of the original data by integrating information from different views. Similarly, clustering algorithms based on Canonical Correlation Analysis (CCA) first project the data from different views into a unified low dimensional subspace, and then apply simple algorithms like single linkage or k -means to achieve the final clustering [33], [34]. Third, unlike the previous methods that try to find a unified representation before applying learning techniques, another strategy in the literature is to integrate the information from individual graphs (views) directly into the optimization problems for the learning purposes. Examples include the co-EM clustering algorithm proposed in [35], and the clustering approaches proposed in [36], [7] based on the frameworks of co-training [37] and co-regularization [38]. Fourth, particularly in the analysis of multiple graphs, regularization frameworks on graphs have also been applied. In [39], the authors have presented a regularization framework over edge weights of multiple graphs to compute an improved similarity graph of the vertices (entities). In [29], [40], the authors have proposed graph regularization frameworks in both vertex and graph spectral domain to combine individual graph layers. Finally, other representative approaches include the works in [41], [39] where the authors have defined additional graph representations to incorporate information from the original individual graphs, and the works in [42], [43], [44], [45] where the authors have proposed ensemble clustering approaches by integrating clustering results from individual views. From this perspective, the proposed approach belongs to the second category mentioned above, where we first find a representative subspace for the information provided by the multi-layer graph and then implement the clustering step, or other learning tasks. We believe that this type of approaches is intuitive and easily understandable, yet still flexible and generic enough to be applied to different types of data.

To summarize, the main differences between the related work and the contributions proposed in this paper are the following. First, the research work on Grassmann manifold theory has been mainly focused on subspace analysis. The subspace usually comes directly from the data but are not linked to graph-based learning problems. Our paper makes the explicit link between subspaces and graphs, and presents a fundamental and intuitive way of approaching the learning problems on multi-layer graphs, with help of subspace analysis on the Grassmann manifold. Second, we show the link between the projection distance on the Grassmann manifold [16], [18] and the empirical estimate of the Hilbert-Schmidt Independence Criterion (HSIC) [5]. Therefore, together with the results in [4], we are able to offer a unified view of concepts from three different perspectives, namely, the projection distance on the Grassmann manifold, the Kullback-Leibler (K-L) divergence [46] and the HSIC [5]. This helps to understand better the key concept of distance measure in subspace analysis. Finally,

using our novel layer merging framework, we provide a simple yet competitive solution to the problem of clustering on multi-layer graphs. We also discuss the influence of the relationships between the individual graph layers on the performance of the proposed clustering algorithm. We believe that this is helpful towards the design of efficient and adaptive learning algorithms.

III. SUBSPACE REPRESENTATION FOR GRAPHS

In this section, we describe a subspace representation for the information provided by a single graph. The subspace representation is inspired by spectral clustering, which studies the spectral properties of the graph information for partitioning the vertex set of the graph into several distinct subsets.

Let us consider an weighted and undirected graph $G = (V, E, \omega)$ ¹, where $V = \{v_i\}_{i=1}^n$ represents the vertex set and E represents the edge set with associated edge weights ω , respectively. Without loss of generality, we assume that the graph is connected. The adjacency matrix W of the graph is a symmetric matrix whose entry W_{ij} represents the edge weight if there is an edge between vertex v_i and v_j , or 0 otherwise. The degree of a vertex is defined as the sum of the weights of all the edges incident to it in the graph, and the degree matrix D is defined as the diagonal matrix containing the degrees of each vertex along its diagonal. The normalized graph Laplacian matrix L is then defined as:

$$L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}. \quad (1)$$

The graph Laplacian is of broad interests in the studies of spectral graph theory [47]. Among several variants, we use the normalized graph Laplacian defined in Eq. (1), since its spectrum (i.e., its eigenvalues) always lie between 0 and 2, a property favorable in comparing different graph layers in the following sections. We consider now the problem of clustering the vertices $V = \{v_i\}_{i=1}^n$ of G into k distinct subsets such that the vertices in the same subset are similar, i.e., they are connected by edges of large weights. This problem can be efficiently solved by the spectral clustering algorithms. Specifically, we focus on the algorithm proposed in [2], which solves the following trace minimization problem:

$$\min_{U \in \mathbb{R}^{n \times k}} \text{tr}(U'LU), \quad \text{s.t. } U'U = I, \quad (2)$$

where n is the number of vertices in the graph, k is the target number of clusters, and $(\cdot)'$ denotes the matrix transpose operator. It can be shown by a version of the Rayleigh-Ritz theorem [3] that the solution U to the problem of Eq. (2) contains the first k eigenvectors (which correspond to the k smallest eigenvalues) of L as columns. The clustering of the vertices in G is then achieved by applying the k -means algorithm [48] to the normalized row vectors of the matrix U^2 . As shown in [3], the behavior of spectral clustering can be explained theoretically with analogies to several well-known mathematical problems, such as the normalized graph-cut problem [1], the random walk process on graphs [49], and

¹We use the notation G for a single graph exclusively in this section.

problems in perturbation theory [50], [51]. This algorithm is summarized in Algorithm 1.

Algorithm 1 Normalized Spectral Clustering [2]

1: Input:

W : the $n \times n$ weighted adjacency matrix of graph G
 k : target number of clusters

2: Compute the degree matrix D and the normalized graph Laplacian matrix $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.

3: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the first k eigenvectors u_1, \dots, u_k of L (solution of (2)). Normalize each row of U to get U_{norm} .

4: Let $y_j \in \mathbb{R}^k$ ($j = 1, \dots, n$) be the j -th row of U_{norm} .

5: Cluster y_j in \mathbb{R}^k into k clusters C_1, \dots, C_k using the k -means algorithm.

6: Output:

C_1, \dots, C_k : the cluster assignment

We provide an illustrative example of the spectral clustering algorithm. Consider a single graph in Fig. 2 (a) with ten vertices that belong to three distinct clusters (i.e., $n=10$ and $k=3$). For the sake of simplicity, all the edge weights are set to 1. The low dimensional matrix U that solves the problem of Eq. (2), which contains k orthonormal eigenvectors of the graph Laplacian L as columns, is shown in Fig. 2 (b). The matrix U is usually called the spectral embedding of the vertices, as each row of U can be viewed as the set of coordinates of the corresponding vertex in the k -dimensional space. More importantly, due to the properties of the graph Laplacian matrix, such an embedding preserves the connectivity of the vertices in the original graph. In other words, two vertices that are strongly connected in the graph are mapped to two vectors (i.e., rows of U) that are close too in the k -dimensional space. As a result, a simple k -means algorithm can be applied to the normalized row vectors of U to achieve the final clustering of the vertices.

Inspired by the spectral clustering theory, one can define a meaningful subspace representation of the original vertices in a graph by its k -dimensional spectral embedding, which is driven by the matrix U built on the first k eigenvectors of the graph Laplacian L . Each row being the coordinates of the corresponding vertex in the low dimensional subspace, this representation contains the information on the connectivity of the vertices in the original graph. Such information can be used for finding clusters of the vertices, as shown above, but it is also useful for other analysis tasks on graphs. By adopting this subspace representation that “summarizes” the graph information, multiple graph layers can naturally be represented by multiple such subspaces (whose geometrical relationships can be quite flexible). The task of multi-layer graph analysis can then be transformed into the problem of effective combination of the multiple subspaces. This is the focus of the next section.

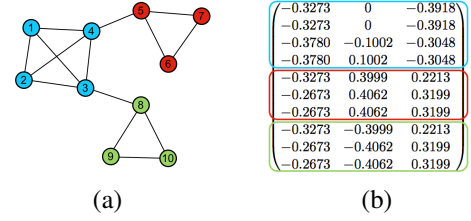


Fig. 2. An illustration of spectral clustering. (a) A graph with three clusters (color-coded) of vertices; (b) Spectral embedding of the vertices computed from the graph Laplacian matrix. The vertices in the same cluster are mapped to coordinates that are close to each other in \mathbb{R}^3 .

IV. MERGING SUBSPACES VIA ANALYSIS ON THE GRASSMANN MANIFOLD

We have described above the subspace representation for each graph layer in the multi-layer graph. We discuss now the problem of effectively combining multiple graph layers by merging multiple subspaces. The theory of Grassmann manifold provides a natural framework for such a problem. In this section, we first review the main ingredients of the Grassmann manifold theory, and then move onto our generic framework for merging subspaces.

A. Ingredients of Grassmann manifold theory

By definition, a Grassmann manifold $\mathcal{G}(k, n)$ is the set of k -dimensional linear subspaces in \mathbb{R}^n , where each unique subspace is mapped to a unique point on the manifold. As an example, Fig. 3 shows two 2-dimensional subspaces in \mathbb{R}^3 being mapped to two points on $\mathcal{G}(2, 3)$. The advantage of using tools from Grassmann manifold theory is thus two-fold: (i) it provides a natural representation for our problem: the subspaces representing the individual graph layers can be considered as different points³ on the Grassmann manifold; (ii) the analysis on the Grassmann manifold permits to use efficient tools to study the distances between points on the manifold, namely, distances between different subspaces. Such distances play an important role in the problem of merging the information from multiple graph layers. In what follows, we focus on the definition of one particular distance measure between subspaces, which will be used in our framework later on.

Mathematically speaking, each point on $\mathcal{G}(k, n)$ can be represented by an orthonormal matrix $Y \in \mathbb{R}^{n \times k}$ whose columns span the corresponding k -dimensional subspace in \mathbb{R}^n ; it is thus denoted as $\text{span}(Y)$. For example, the two subspaces shown in Fig. 3 can be denoted as $\text{span}(Y_1)$ and $\text{span}(Y_2)$ for two orthonormal matrices Y_1 and Y_2 . The distance between two points on the manifold, or between two subspaces $\text{span}(Y_1)$ and $\text{span}(Y_2)$, is then defined based on a set of principal angles $\{\theta_i\}_{i=1}^k$ between these subspaces [52]. These principal angles, which measure how the subspaces are geometrically close, are the fundamental measures used to define various distances on the Grassmann manifold, such as

²The necessity for row normalization is discussed in [3] and we omit this discussion here. However, the normalization does not change the nature of spectral embedding, hence, it does not affect our derivation later.

³We assume that the Laplacian matrices of any pair of the two layers in the multi-layer graph have different sets of top eigenvectors. In this case, subspace representations for all the layers will be different from each other.

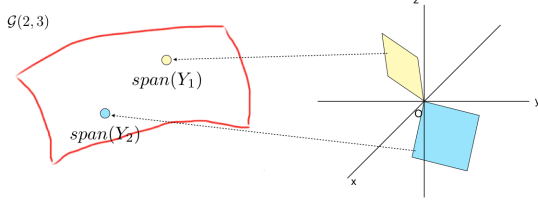


Fig. 3. An example of two 2-dimensional subspaces $\text{span}(Y_1)$ and $\text{span}(Y_2)$ in \mathbb{R}^3 , which are mapped to two points on the Grassmann manifold $\mathcal{G}(2,3)$.

the Riemannian (geodesic) distance or the projection distance [16], [18]. In this paper, we use the projection distance, which is defined as:

$$d_{\text{proj}}(Y_1, Y_2) = \left(\sum_{i=1}^k \sin^2 \theta_i \right)^{\frac{1}{2}}, \quad (3)$$

where Y_1 and Y_2 are the orthonormal matrices representing the two subspaces under comparison⁴. The reason for choosing the projection distance is two-fold: (i) the projection distance is defined as the ℓ^2 -norm of the vector of sines of the principal angles. Since it uses all the principal angles, it is therefore an unbiased definition. This is favorable as we do not assume any prior knowledge on the distribution of the data, and all the principal angles are considered to carry meaningful information; (ii) the projection distance can be interpreted using a one-to-one mapping that preserves distinctness: $\text{span}(Y) \rightarrow YY' \in \mathbb{R}^{n \times n}$. Note that the squared projection distance can be rewritten as:

$$\begin{aligned} d_{\text{proj}}^2(Y_1, Y_2) &= \sum_{i=1}^k \sin^2 \theta_i \\ &= k - \sum_{i=1}^k \cos^2 \theta_i \\ &= k - \text{tr}(Y_1 Y_1' Y_2 Y_2') \\ &= \frac{1}{2} [2k - 2\text{tr}(Y_1 Y_1' Y_2 Y_2')] \\ &= \frac{1}{2} [\text{tr}(Y_1' Y_1) + \text{tr}(Y_2' Y_2) - 2\text{tr}(Y_1 Y_1' Y_2 Y_2')] \\ &= \frac{1}{2} \|Y_1 Y_1' - Y_2 Y_2'\|_F^2, \end{aligned} \quad (4)$$

where the third equality comes from the definition of the principal angles and the fifth equality uses the fact that Y_1 and Y_2 are orthonormal matrices. It can be seen from Eq. (4) that the projection distance can be related to the Frobenius norm of the difference between the mappings of the two subspaces $\text{span}(Y_1)$ and $\text{span}(Y_2)$ in $\mathbb{R}^{n \times n}$. Because the mapping preserves distinctness, it is natural to take the projection distance as a proper distance measure between subspaces. Moreover, the third equality of Eq. (4) provides an explicit way of computing the projection distance between two subspaces from their matrix representations Y_1 and Y_2 . We are

going to use it in developing the generic merging framework in the following section.

To summarize, the Grassmann manifold provides a natural and intuitive representation for subspace-based analysis (as shown in Fig. 3). The associated tools, namely the principal angles, permit to define a meaningful distance measure that captures the geometric relationships between the subspaces. Originally defined as a distance measure between two subspaces, the projection distance can be naturally generalized to the analysis of multiple subspaces, as we show in the next section.

B. Generic merging framework

Equipped with the subspace representation for individual graphs and with a distance measure to compare different subspaces, we are now ready to present our generic framework for merging the information from multiple graph layers. Given a multi-layer graph G with M individual layers $\{G_i\}_{i=1}^M$, we first compute the graph Laplacian matrix L_i for each G_i and then represent each G_i by the spectral embedding matrix $U_i \in \mathbb{R}^{n \times k}$ from the first k eigenvectors of L_i , where n is the number of vertices and k is the target number of clusters. Recall that each of the matrices $\{U_i\}_{i=1}^M$ defines a k -dimensional subspace in \mathbb{R}^n , which can be denoted as $\text{span}(U_i)$. The goal is to merge these multiple subspaces in a meaningful and efficient way. To this end, our philosophy is to find a representative subspace $\text{span}(U)$ that is close to all the individual subspaces $\text{span}(U_i)$, and at the same time the representation U preserves the vertex connectivity in each graph layer. For notational convenience, in the rest of the paper we simply refer to the representations U and U_i as the corresponding subspaces, unless indicated specifically.

The squared projection distance between subspaces defined in Eq. (4) can be naturally generalized for analysis of multiple subspaces. More specifically, we can define the squared projection distance between the target representative subspace U and the M individual subspaces $\{U_i\}_{i=1}^M$ as the sum of squared projection distances between U and each individual subspace given by U_i :

$$\begin{aligned} d_{\text{proj}}^2(U, \{U_i\}_{i=1}^M) &= \sum_{i=1}^M d_p^2(U, U_i) \\ &= \sum_{i=1}^M [k - \text{tr}(UU'U_iU_i')] \\ &= kM - \sum_{i=1}^M \text{tr}(UU'U_iU_i'). \end{aligned} \quad (5)$$

The minimization of the distance measure in Eq. (5) enforces the representative subspace U to be close to all the individual subspaces $\{U_i\}_{i=1}^M$ in terms of the projection distance on the Grassmann manifold. At the same time, we want U to preserve the vertex connectivity in each graph layer. This can be achieved by minimizing the Laplacian quadratic form evaluated on the columns of U , as also indicated by the objective function in Eq. (2) for spectral clustering. Therefore, we finally propose to merge multiple subspaces by solving the

⁴In the special case where Y_1 and Y_2 represent the same subspace, we have $d_{\text{proj}}(Y_1, Y_2) = 0$.

following optimization problem that integrates Eq. (2) and Eq. (5):

$$\min_{U \in \mathbb{R}^{n \times k}} \sum_{i=1}^M \text{tr}(U' L_i U) + \alpha [kM - \sum_{i=1}^M \text{tr}(U U' U_i U_i')], \quad (6)$$

$$\text{s.t. } U' U = I,$$

where L_i and U_i are the graph Laplacian and the subspace representation for G_i , respectively. The regularization parameter α balances the trade-off between the two terms in the objective function.

The problem of Eq. (6) can be solved in a similar manner as Eq. (2). Specifically, by ignoring constant terms and rearranging the trace form in the second term of the objective function, Eq. (6) can be rewritten as

$$\min_{U \in \mathbb{R}^{n \times k}} \text{tr}[U' (\sum_{i=1}^M L_i - \alpha \sum_{i=1}^M U_i U_i') U], \quad \text{s.t. } U' U = I. \quad (7)$$

It is interesting to note that this is the same trace minimization problem as in Eq. (2), but with a “modified” Laplacian:

$$L_{\text{mod}} = \sum_{i=1}^M L_i - \alpha \sum_{i=1}^M U_i U_i'. \quad (8)$$

Therefore, by the Rayleigh-Ritz theorem, the solution to the problem of Eq. (7) is given by the first k eigenvectors of the modified Laplacian L_{mod} , which can be computed using efficient algorithms for eigenvalue problems [53], [54].

In the problem of Eq. (6) we try to find a representative subspace U from the multiple subspaces $\{U_i\}_{i=1}^M$. Such a representation not only preserves the structural information contained in the individual graph layers, which is encouraged by the first term of the objective function in Eq. (6), but also keeps a minimum distance between itself and the multiple subspaces, which is enforced by the second term. Notice that the minimization of only the first term itself corresponds to simple averaging of the information from different graph layers, which usually leads to suboptimal clustering performance as we shall see in the experimental section. Similarly, imposing only a small projection distance to the individual subspaces $\{U_i\}_{i=1}^M$ does not necessarily guarantee that U is a good solution for merging the subspaces. In fact, for a given k -dimensional subspace, there are infinitely many choices for the matrix representation, and not all of them are considered as meaningful summarizations of the information provided by the multiple graph layers. However, under the additional constraint of minimizing the trace of the quadratic term $U' L_i U$ over all the graphs (which is the first term of the objective function in Eq. (6)), the vertex connectivity in the individual graphs tends to be preserved in U . In this case, the smaller the projection distance between U and the individual subspaces, the more representative it is for all graph layers.

C. Discussion of the distance function

Interestingly, the choice of projection distance as a similarity measure between subspaces in the optimization problem of Eq. (6) can be well justified from information-theoretic and

statistical learning points of view. The first justification is from the work of Hamm et al. [4], in which the authors have shown that the Kullback-Leibler (K-L) divergence [46], which is a well-known similarity measure between two probability distributions in information theory, is closely related to the squared projection distance. More specifically, the work in [4] suggests that, under certain conditions, we can consider a linear subspace U_i as the “flattened” limit of a Factor Analyzer distribution p_i [55]:

$$p_i : \mathcal{N}(u_i, C_i), \quad C_i = U_i U_i' + \sigma^2 I_D, \quad (9)$$

where \mathcal{N} stands for the normal distribution, $u_i \in \mathbb{R}^n$ is the mean, $U_i \in \mathbb{R}^{n \times k}$ is a full-rank matrix with $n > k > 0$ (which represents the subspace), σ is the ambient noise level, and I_n is the identity matrix of dimension n . For two subspaces U_i and U_j , the symmetrized K-L divergence between the two corresponding distributions p_i and p_j can then be rewritten as:

$$d_{\text{KL}}(p_1, p_2) = \frac{1}{2\sigma^2(\sigma^2 + 1)} (2k - 2\text{tr}(U_i U_i' U_j U_j')), \quad (10)$$

which is of the same form as the squared projection distance when we ignore the constant factor (see Eq. (4)). This shows that, if we take a probabilistic view of the subspace representations $\{U_i\}_{i=1}^M$, then the projection distance between subspaces can be considered consistent with the K-L divergence.

The second justification is from the recently proposed Hilbert-Schmidt Independence Criterion (HSIC) [5], which measures the statistical dependence between two random variables. Given $K_{\mathcal{X}_1}, K_{\mathcal{X}_2} \in \mathbb{R}^{n \times n}$ that are the centered Gram matrices of some kernel functions defined over two random variables \mathcal{X}_1 and \mathcal{X}_2 , the empirical estimate of HSIC is given by

$$d_{\text{HSIC}}(\mathcal{X}_1, \mathcal{X}_2) = \text{tr}(K_{\mathcal{X}_1} K_{\mathcal{X}_2}). \quad (11)$$

That is, the larger the $d_{\text{HSIC}}(\mathcal{X}_1, \mathcal{X}_2)$, the stronger the statistical dependence between \mathcal{X}_1 and \mathcal{X}_2 . In our case, using the idea of spectral embedding, we can consider the rows of the individual subspace representations U_i and U_j as two particular sets of sample points in \mathbb{R}^k , which are drawn from two probability distributions governed by the information on vertex connectivity in G_i and G_j , respectively. In other words, the sets of rows of U_i and U_j can be seen as realizations of two random variables \mathcal{X}_i and \mathcal{X}_j . Therefore, we can define the Gram matrices of linear kernels on \mathcal{X}_i and \mathcal{X}_j as:

$$K_{\mathcal{X}_i} = (U_i')'(U_i') = U_i U_i',$$

$$K_{\mathcal{X}_j} = (U_j')'(U_j') = U_j U_j'. \quad (12)$$

By applying Eq. (11), we can see that:

$$d_{\text{HSIC}}(\mathcal{X}_i, \mathcal{X}_j) = \text{tr}(K_{\mathcal{X}_i} K_{\mathcal{X}_j})$$

$$= \text{tr}(U_i U_i' U_j U_j')$$

$$= k - d_{\text{proj}}^2(U_i, U_j). \quad (13)$$

This shows that the projection distance between subspaces U_i and U_j can be interpreted as the negative dependence between \mathcal{X}_i and \mathcal{X}_j , which reflect the information provided by the two individual graph layers G_i and G_j .

Therefore, from both information-theoretic and statistical learning points of view, the smaller the projection distance between two subspace representations U_i and U_j , the more similar the information in the respective graphs that they represent. As a result, the representative subspace (the solution U to the problem of Eq. (6)) can be considered as a subspace representation that “summarizes” the information from the individual graph layers, and at the same time captures the intrinsic relationships between the vertices in the graph. As one can imagine, such relationships are of crucial importance in our multi-layer graph analysis.

In summary, the concept of treating individual graphs as subspaces, or points on the Grassmann manifold, permits to study the desired merging framework in a unique and principled way. We are able to find a representative subspace for the multi-layer graph of interest, which can be viewed as a dimensionality reduction approach for the original data. We finally remark that the proposed merging framework can be easily extended to take into account the relative importance of each individual graph layer with respect to the specific learning purpose. For instance, when prior knowledge about the importance of the information in the individual graphs is available, we can adapt the value of the regularization parameter α in Eq. (6) to the different layers such that the representative subspace is closer to the most informative subspace representations.

V. CLUSTERING ON MULTI-LAYER GRAPHS

In Section IV, we introduced a novel framework for merging subspace representations from the individual layers of a multi-layer graph, which leads to a representative subspace that captures the intrinsic relationships between the vertices of the graph. This representative subspace provides a low dimensional form that can be used in several applications involving multi-layer graph analysis. In particular, we study now one such application, namely the problem of clustering vertices in a multi-layer graph. We further analyze the behavior of the proposed clustering algorithm with respect to the properties of the individual graph layers (subspaces).

A. Clustering algorithm

As we have already seen in Section III, the success of the spectral clustering algorithm relies on the transformation of the information contained in the graph structure into a spectral embedding computed from the graph Laplacian matrix, where each row of the embedding matrix (after normalization) is treated as the coordinates of the corresponding vertex in a low dimensional subspace. In our problem of clustering on a multi-layer graph, the setting is slightly different, since we aim at finding a unified clustering of the vertices that takes into account information contained in all the individual layers of the multi-layer graph. However, the merging framework proposed in the previous section can naturally be applied in this context. In fact, it leads to a natural solution to the clustering problem on multi-layer graphs. In more details, similarly to the spectral embedding matrix in the spectral clustering algorithm, which is a subspace representation for

one individual graph, our merging framework provides a representative subspace that contains the information from the multiple graph layers. Using this representation, we can then follow the same steps of spectral clustering to achieve the final clustering of the vertices with a k -means algorithm. The proposed clustering algorithm is summarized in Algorithm 2.

Algorithm 2 Spectral Clustering on Multi-Layer graphs (SC-ML)

- 1: **Input:**
 $\{W_i\}_{i=1}^M$: $n \times n$ weighted adjacency matrices of individual graph layers $\{G_i\}_{i=1}^M$
 k : target number of clusters
 α : regularization parameter
 - 2: Compute the normalized Laplacian matrix L_i and the subspace representation U_i for each G_i .
 - 3: Compute the modified Laplacian matrix $L_{\text{mod}} = \sum_{i=1}^M L_i - \alpha \sum_{i=1}^M U_i U_i'$.
 - 4: Compute $U \in \mathbb{R}^{n \times k}$ that is the matrix containing the first k eigenvectors u_1, \dots, u_k of L_{mod} . Normalize each row of U to get U_{norm} .
 - 5: Let $y_j \in \mathbb{R}^k$ ($j = 1, \dots, n$) be the j -th row of U_{norm} .
 - 6: Cluster y_j in \mathbb{R}^k into C_1, \dots, C_k using the k -means algorithm.
 - 7: **Output:**
 C_1, \dots, C_k : The cluster assignment
-

It is clear that Algorithm 2 is a direct generalization of Algorithm 1 in the case of multi-layer graphs. The main ingredient of our clustering algorithm is the merging framework proposed in Section IV, in which information from individual graph layers is summarized, prior to the actual clustering process (i.e., the k -means step) is implemented. This provides an example that illustrates how our generic merging framework can be applied to specific learning tasks on multi-layer graphs.

B. Analysis of the proposed algorithm

We now analyze the behavior of the proposed clustering algorithm under different conditions. Specifically, we first outline the link between subspace distance and clustering quality, and then compare the clustering performances in two scenarios where the relationships between the individual subspaces $\{U_i\}_{i=1}^M$ are different.

As we have seen in Section IV, the rows of the subspace representations $\{U_i\}_{i=1}^M$ can be viewed as realizations of random variables $\{\mathcal{X}_i\}_{i=1}^M$ governed by the graph information. At the same time, spectral clustering directly utilizes U_i for the purpose of clustering. Therefore, $\{\mathcal{X}_i\}_{i=1}^M$ can be considered as random variables that control the cluster assignment of the vertices. In fact, it has been shown in [3] that the matrix U_i is closely related to the matrix that contains the cluster indicator vectors as columns. Since the projection distance can be understood as the negative statistical dependence between such random variables, the minimization of the projection distance in Eq. (6) is equivalent to the maximization of the dependence between the random variable from the representative subspace U and the ones from the individual subspaces $\{U_i\}_{i=1}^M$. The

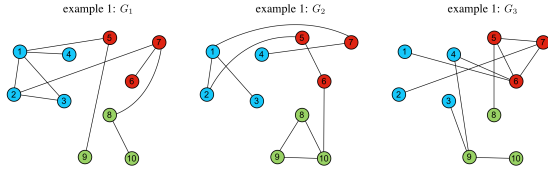


Fig. 4. A 3-layer graph with unit edge weights for toy example 1. The colors indicate the groundtruth clusters.

TABLE I
ANALYSIS OF TOY EXAMPLE 1.

	layer G_1	layer G_2	layer G_3	SC-ML
NMI	0.6279	0.6181	0.2673	1.0000

(a) clustering performances for toy example 1

	layer G_1	layer G_2	layer G_3	subspace computed by SC-ML
layer G_1	0	1.1100	1.3670	0.9456
layer G_2	1.1100	0	1.3354	1.0452
layer G_3	1.3670	1.3354	0	1.0788

(b) subspace distances for toy example 1

optimization in Eq. (6) can then be seen as a solution that tends to produce a clustering with the representative subspace that is consistent with those computed from the individual subspace representations.

We now discuss how the relationships between the individual subspaces possibly affect the performance of our clustering algorithm **SC-ML**. Intuitively, since the second term of the objective function in Eq. (6) represents the distance between the representative subspace U and all the individual subspaces $\{U_i\}_{i=1}^M$, it tends to drive the solution towards those subspaces that themselves are close to each other on the Grassmann manifold. To show it more clearly, let us consider two toy examples. The first example is illustrated in Fig. 4, where we have a 3-layer graph with the individual layers G_1 , G_2 and G_3 sharing the same set of vertices. For the sake of simplicity, all the edge weights are set to one. In addition, three groundtruth clusters are indicated by the colors of the vertices. Table I (a) shows the performances of Algorithm 1 with individual layers as well as Algorithm 2⁵ for the multi-layer graph, in terms of *Normalized Mutual Information* (NMI) [56] with respect to the groundtruth clusters. Table I (b) shows the projection distances between various pairs of subspaces. It is clear that the layers G_1 and G_2 produce better clustering quality, and that the distance between the corresponding subspaces is smaller. However, the vertex connectivity in layer G_3 is not very consistent with the groundtruth clusters and the corresponding subspace is further away from the ones from G_1 and G_2 . In this case, the solution found by **SC-ML** is enforced to be close to the consistent subspaces from G_1 and G_2 , hence provides satisfactory clustering results ($NMI = 1$ represents perfect recovery of groundtruth clusters). Let us now consider a second toy example, as illustrated in Fig. 5. In this example we have two layers G_2 and G_3 with relatively low quality information with respect to the groundtruth clustering of the vertices. As we see in Table II (b), their corresponding

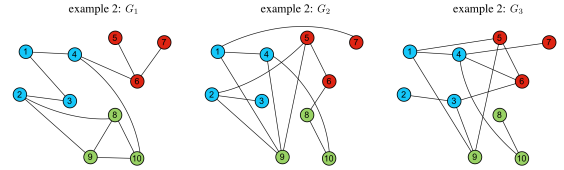


Fig. 5. A 3-layer graph with unit edge weights for toy example 2. The colors indicate the groundtruth clusters.

TABLE II
ANALYSIS OF TOY EXAMPLE 2.

	layer G_1	layer G_2	layer G_3	SC-ML
NMI	0.7934	0.2673	0.4728	0.5300

(a) clustering performances for toy example 2

	layer G_1	layer G_2	layer G_3	subspace computed by SC-ML
layer G_1	0	1.3098	1.2296	1.0311
layer G_2	1.3098	0	0.9343	0.8828
layer G_3	1.2296	0.9343	0	0.5058

(b) subspace distances for toy example 2

subspaces are close to each other on the Grassmann manifold. The most informative layer G_1 , however, represents a subspace that is quite far away from the ones from G_2 and G_3 . At the same time, we see in Table II (a) that the clustering results are better for the first layer than for the other two less informative layers. If the quality of the information in the different layers is not considered in computing the representative subspace, **SC-ML** enforces the solution to be closer to two layers of relatively lower quality, which results in unsatisfactory clustering performance in this case.

The analysis above implies that the proposed clustering algorithm works well under the following assumptions: (i) the majority of the individual subspaces are relatively informative, namely, they are helpful for recovering the groundtruth clustering, and (ii) they are reasonably close to each other on the Grassmann manifold, namely, they provide *complementary* but not *contradictory* information. These are the assumptions made in the present work. As we shall see in the next section, these assumptions seem to be appropriate and realistic in real world datasets. If it is not the case, one may assume that a preprocessing step cleans the datasets, or at least provides information about the reliability of the information in the different graph layers.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the **SC-ML** algorithm presented in Section V on several synthetic and real world datasets. We first describe the datasets that we use for the evaluation, and then explain the various clustering algorithms that we adopt in the performance comparisons. We finally present the results in terms of three evaluation criteria as well as some discussions.

⁵We choose the value of the regularization parameter that leads to the best possible clustering performance. More discussions about the choices of this parameter are presented in Section VI.

A. Datasets

We adopt one synthetic and two real world datasets with multi-layer graph representation for the evaluation of the clustering algorithms. We give a brief overview of the datasets as follows.

The first dataset that we use is a synthetic dataset, where we have three point clouds in \mathbb{R}^2 forming the English letters “N”, “R” and “C” (shown in Fig. 6). Each point cloud is generated from a five-component Gaussian mixture model with different values for the mean and variance of the Gaussian distributions, where each component represents a class of 500 points with specific color. A 5-nearest neighbor graph is then constructed for each point cloud by assigning the weight of the edges connecting two vertices (points) as the reciprocal of the Euclidean distance between them. This gives us a 3-layer graph of 2500 vertices, where each graph layer is from a point cloud forming a particular letter. The goal with this dataset is to recover the five clusters (indicated by five colors) of the 2500 vertices using the three graph layers constructed from the three point clouds.

The second dataset contains data collected during the Lausanne Data Collection Campaign [57] by the Nokia Research Center (NRC) in Lausanne. This dataset contains the mobile phone data of 136 users living and working in the Lake Léman region in Switzerland, recorded over a one-year period. Considering the users as vertices in the graph, we construct three graphs by measuring the proximities between these users in terms of GPS locations, Bluetooth scanning activities and phone communication. More specifically, for GPS locations and bluetooth scans, we measure how many times two users are sufficiently close geographically (within a distance of roughly 1 km), and how many times two users’ devices have detected the same bluetooth devices, respectively, within 30-minute time windows. Aggregating these results for a one-year period leads to two weighted adjacency matrices that represent the physical proximities of the users measured with different modalities. In addition, an adjacency matrix for phone communication is generated by assigning edge weights depending on the number of calls between any pair of two users. These three adjacency matrices form a 3-layer graph of 136 vertices, where the goal is to recover the eight groundtruth clusters that have been constructed from the users’ email affiliations.

The third dataset is a subset of the Cora bibliographic dataset⁶. This dataset contains 292 research papers from three different fields, namely, natural language processing, data mining and robotics. Considering papers as vertices in the graph, we construct the first two graphs by measuring the similarities among the title and the abstract of these papers. More clearly, for both title and abstract, we represent each paper by a vector of non-trivial words using the *Term Frequency-Inverse Document Frequency (TF-IDF)* weighting scheme, and compute the cosine similarities between every pair of vectors as the edge weights in the graphs. Moreover, we add a third graph which reflects the citation relationships among the papers, namely, we assign an edge with unit weight between papers A and B if A has cited or been cited by B .

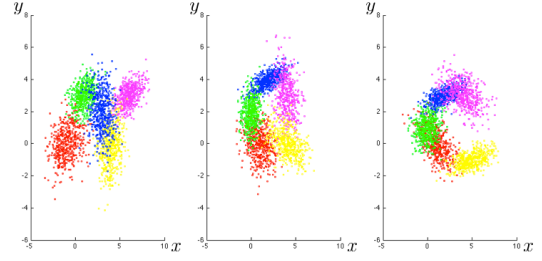


Fig. 6. Three five-class point clouds in \mathbb{R}^2 forming English letters “N”, “R” and “C”.

This results in a 3-layer graph of 292 vertices, and the goal in this dataset is to recover the three clusters corresponding to the different fields the papers belong to.

To visualize the graphs in the three datasets, the spy plot of the adjacency matrices of the graphs are shown in Fig. 7 (a), (b) and (c) for the synthetic, NRC and Cora dataset, respectively, where the orderings of the vertices are made consistent with the groundtruth clusters⁷. A spy plot is a global view of a matrix where every non-zero entry in the matrix is represented by a blue dot (without taking into account the value of the entry). As shown in these figures, we see clearly the clusters in the synthetic and Cora datasets, while the clusters in the NRC dataset are not very clear. The reason for this is that, in the NRC dataset, the email affiliations used to create the groundtruth clusters only provides approximative information.

B. Clustering algorithms

We now explain briefly the clustering algorithms in our comparative performance analysis along with some implementation details. We adopt three baseline algorithms as well as a state-of-the-art technique, namely the co-regularization approach introduced in [7]. As we shall see, there is an interesting connection between this approach and the proposed algorithm. First of all, we describe some implementation details of the proposed **SC-ML** algorithm and the co-regularization approach in [7]:

- **SC-ML**: Spectral Clustering on Multi-Layer graphs, as presented in Section V. The implementation of **SC-ML** is pretty straightforward, and the only parameter to choose is the regularization parameter α in Eq. (6). In our experiments, we choose the value of α through multiple empirical trials and report the best clustering performance. Specifically, we choose α to be 0.64 for the synthetic dataset and 0.44 for both real world datasets. We will discuss the choice of this parameter later in this section.
- **SC-CoR**: Spectral Clustering with Co-Regularization proposed in [7]. We follow the same practice as in [7] to choose the most informative graph layer to initialize

⁶Available online at “<http://people.cs.umass.edu/~mccallum/data.html>” under category “Cora Research Paper Classification”.

⁷The adjacency matrix for GPS proximity in the NRC dataset is thresholded for better illustration.

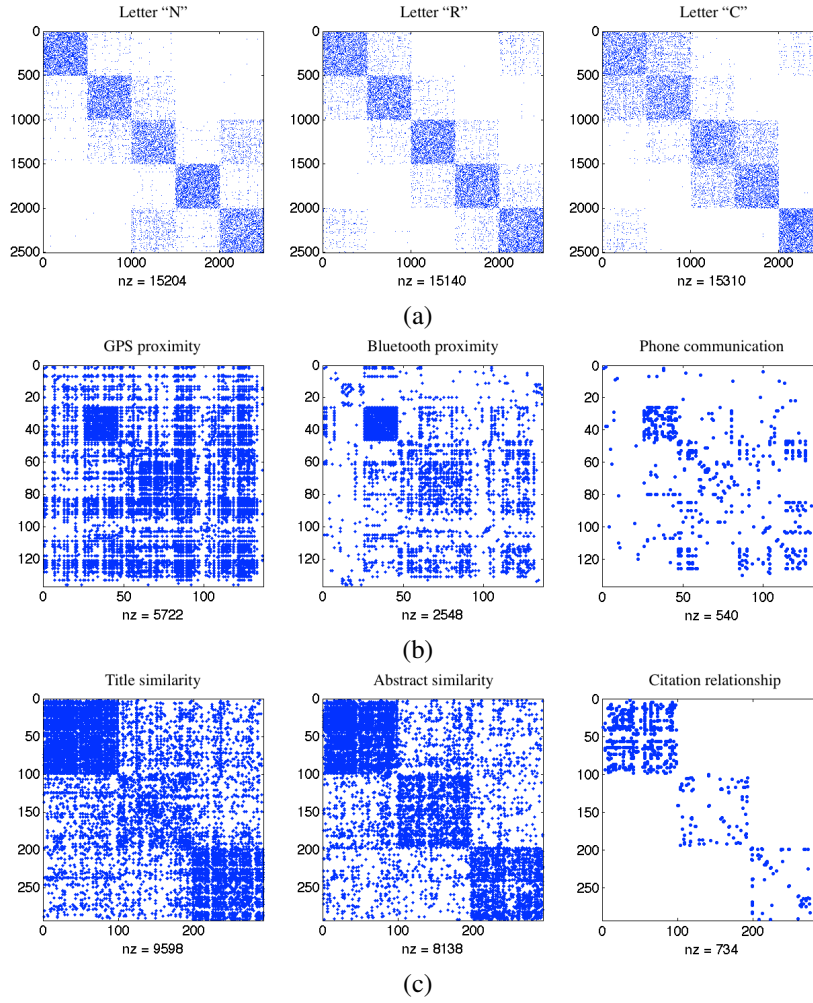


Fig. 7. Spy plots of three adjacency matrices in (a) the synthetic dataset, (b) the NRC dataset, and (c) the Cora dataset.

the alternating optimization scheme in **SC-CoR**. The stopping criteria for the optimization process is chosen such that the optimization stops when changes in the objective function are smaller than 10^{-5} . Similarly, we choose the value of the regularization parameter α in **SC-CoR** through multiple empirical trials and report the best clustering performance. As in [7], the parameter α is fixed in the optimization steps for all graph layers.

Next, we introduce three baseline comparative algorithms that work as follows:

- **SC-Single**: Spectral Clustering (Algorithm 1) applied on a single graph layer, where the graph is chosen to be the one that leads to the best clustering results.
- **SC-Sum**: Spectral clustering applied on a global matrix W that is the summation of the normalized adjacency matrices of the individual layers:

$$W = \sum_{i=1}^M D_i^{-\frac{1}{2}} W_i D_i^{-\frac{1}{2}}. \quad (14)$$

- **SC-KSum**: Spectral clustering applied on the summation

K of the spectral kernels [6] of the adjacency matrices:

$$K = \sum_{i=1}^M K_i \quad \text{with} \quad K_i = \sum_{m=1}^d u_{im} u_{im}', \quad (15)$$

where n is the number of vertices, $d \ll n$ is the number of eigenvectors used in the definition of the spectral kernels K_i , and u_{im} represents the m -th eigenvector of the Laplacian L_i for graph G_i . To make it more comparable with spectral clustering, we choose d to be the target number of clusters in our experiments.

C. Results and discussions

We evaluate the performance of the different clustering algorithms with three different criteria, namely *Purity*, *Normalized Mutual Information (NMI)* and *Rand Index (RI)* [56]. The results are summarized in Table III (a), (b) and (c) for the synthetic, NRC and Cora dataset, respectively. For each scenario, the best two results are highlighted in bold fonts. First, as expected, we see that the clustering performances for the synthetic and Cora datasets are higher than that for the NRC dataset, which indicates that the latter one is indeed more challenging due to the approximative groundtruth information. Second, it is clear that **SC-ML** and **SC-CoR**

TABLE III

PERFORMANCE COMPARISON OF DIFFERENT CLUSTERING ALGORITHMS ON (A) THE SYNTHETIC DATASET, (B) THE NRC DATASET, AND (C) THE CORA DATASET.

	SC-Single	SC-Sum	SC-KSum	SC-CoR	SC-ML
<i>Purity</i>	0.8580	0.9752	0.9768	0.9784	0.9828
<i>NMI</i>	0.7266	0.9224	0.9262	0.9278	0.9407
<i>RI</i>	0.9018	0.9806	0.9818	0.9830	0.9864

(a)

	SC-Single	SC-Sum	SC-KSum	SC-CoR	SC-ML
<i>Purity</i>	0.5147	0.5956	0.5294	0.5809	0.6103
<i>NMI</i>	0.3133	0.3988	0.3440	0.4056	0.4156
<i>RI</i>	0.7326	0.7852	0.7667	0.7878	0.7929

(b)

	SC-Single	SC-Sum	SC-KSum	SC-CoR	SC-ML
<i>Purity</i>	0.9555	0.9795	0.9726	0.9829	0.9829
<i>NMI</i>	0.8314	0.9062	0.8863	0.9175	0.9175
<i>RI</i>	0.9426	0.9731	0.9645	0.9775	0.9775

(c)

generally outperform the baseline approaches for the three datasets. More specifically, although both **SC-Sum** and **SC-KSum** indeed improve the clustering quality compared to clustering with individual graph layers, they only provide limited improvement, and the potential drawback for both of the summation methods is that they can be considered as similar to building a simple average graph for representing the different layers of information. Therefore, depending on data characteristics in specific datasets, this might smooth out the particular information provided by individual layers, and thus penalize the clustering performance. In comparison, **SC-ML** and **SC-CoR** always achieve significant improvements in the clustering quality compared to clustering using individual graph layers.

We now take a closer look at the comparisons between **SC-ML** and **SC-CoR**. Although the latter is not developed from the viewpoint of subspace analysis on the Grassmann manifold, it can actually be interpreted as a process in which individual subspace representations are updated based on the same distance analysis as in our framework. In this sense, **SC-CoR** uses the same distance as ours to measure similarities between subspaces. The merging solution however leads to a different optimization problem than that of Eq. (6), which is based on a slightly different merging philosophy. Specifically, it enforces the information contained in the individual subspace representations to be consistent with each other. An alternating optimization scheme optimizes, at each step, one subspace representation, while fixing the others. This can be interpreted as a process in which one subspace at each step becomes closer to other subspaces in term of the projection distance on the Grassmann manifold. Upon convergence, all initial subspaces are “brought” closer

to each other and the final subspace representation from the most informative graph layer is considered as the one that combines information from all the graph layers efficiently. Two illustrations of **SC-CoR** and **SC-ML** are shown in Fig. 8 (a) and (b), respectively. Therefore, on the one hand, results for both approaches demonstrate the benefit of using our distance analysis on the Grassmann manifold for merging information in multi-layer graphs. Indeed, for both approaches, since the distances between the solutions and the individual subspaces are minimized without sacrificing too much of the information from individual graph layers, the resulting combinations can be considered as good summarizations of the multiple graph layers. On the other hand, however, **SC-ML** differs from **SC-CoR** mainly in the following aspects. First, the alternating optimization scheme in **SC-CoR** focuses only on optimizing one subspace representation at each step, and it requires a sensible initialization to guarantee that the algorithm ends up at a good local minimum for the optimization problem; it also does not guarantee that all the subspace representations converge to one point on the Grassmann manifold (it uses the final update of the most informative layer for clustering)⁸. In contrast, **SC-ML** directly finds a single representation through a unique optimization of the representative subspace with respect to all graph layers jointly, which does not need alternating optimization steps and careful initializations. These are the possible reasons that explain why **SC-ML** performs better than **SC-CoR** in our experiments, as we can see in Table III. Second, it is worth noting that, from a computational point of view, the optimization process involved in **SC-ML** is much simpler than that in **SC-CoR**. Specifically, the iterative nature of **SC-CoR** requires solving an eigenvalue problem for

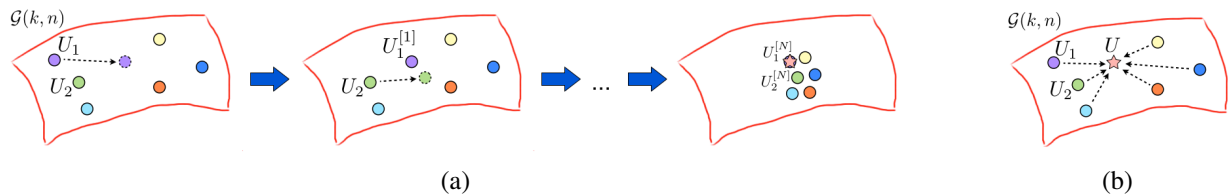


Fig. 8. Illustrations of graph layer merging. (a) Co-regularization [7]: iterative update of the individual subspace representations. The upper index $[N]$ represents the number of iterative steps on each individual subspace representation. The final update of the subspace representation for the most informative graph ($U_1^{[N]}$, shown as a star) is considered as a good combination; (b) Proposed merging framework: the representative subspace (U , shown as a star) is found in one step.

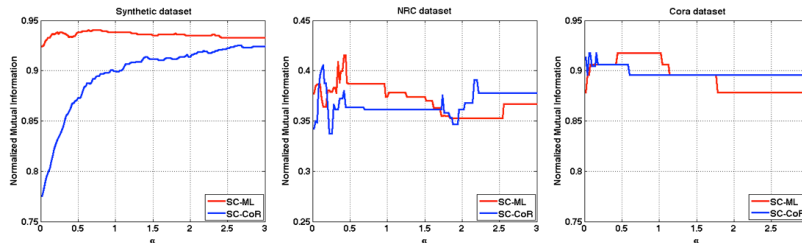


Fig. 9. Performances of **SC-ML** and **SC-CoR** under different values of parameter α in the corresponding implementations.

MN times, where M and N are the number of individual graphs and the number of iterations needed for the algorithm to converge, respectively. In contrast, since **SC-ML** aims at finding a globally representative subspace without modifying the individual ones, it needs to solve an eigenvalue problem only once.

Finally, we discuss the influence of the choice of the regularization parameter α on the performance of **SC-ML**. In Fig. 9, we compare the performances of **SC-ML** and **SC-CoR** in terms of *NMI* under different values of parameter α in the corresponding implementations. As we can see, in our experiments, **SC-ML** achieves the best performances when α is chosen between 0.4 and 0.6, and it outperforms **SC-CoR** for a large range of α for the synthetic and NRC datasets. For the Cora dataset, the two algorithms achieve the same performance at different values of α , but **SC-ML** permits a larger range of parameter selection. Furthermore, it is worth noting that the optimal values for α in **SC-ML** lie in similar ranges across different datasets, thanks to the adoption of the normalized graph Laplacian matrix whose spectral norm is upper bounded by 2. In summary, this shows that the performance of **SC-ML** is reasonably stable with respect to the parameter selection.

VII. CONCLUSIONS

In this paper, we provide a framework for analyzing information provided by multi-layer graphs and for clustering vertices of graphs in rich datasets. Our generic approach is based on the transformation of information contained in the individual graph layers into subspaces on the Grassmann manifold. The estimation of a representative subspace can then be essentially considered as the problem of finding a good

⁸In [7], the authors have also proposed a “centroid-based co-regularization approach” that introduces a consensus representation. However, such a representation is still computed via an alternating optimization scheme, which needs a sensible initialization and keeps the same iterative nature.

summarization of multiple subspaces using distance analysis on the Grassmann manifold. The proposed approach can be applied to various learning tasks where multiple subspace representations are involved. Under appropriate and realistic assumptions, we show that our framework can be applied to the clustering problem on multi-layer graphs and that it provides an efficient solution that is competitive to the state-of-the-art techniques. Finally, we mention the following research directions as interesting and open problems. First, the subspace representation inspired by spectral clustering is not the only valid representation for the graph information. As suggested by the works in [10], [11], the eigenvectors of the modularity matrix of the graph can also be used as low dimensional subspace representation for the information contained in the graph. Therefore, an interesting problem is to find the most appropriate subspace representation for the data available, either they are graphs or of some more general forms. Second, we believe that better clustering performance can be achieved if prior information on the data is available, in particular about the consistency of the information in the different graph layers. These problems are however left for future studies.

VIII. ACKNOWLEDGEMENT

This work has been partly supported by Nokia Research Center (NRC) Lausanne, and the EDGAR project funded by Hasler Foundation, Switzerland.

REFERENCES

- [1] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.
- [2] A. Ng, M. Jordan, and Y. Weiss, “On Spectral Clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems (NIPS)*, 2002.
- [3] U. von Luxburg, “A Tutorial on Spectral Clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec 2007.

- [4] J. Hamm and D. Lee, "Extended Grassmann Kernels for Subspace-Based Learning," *Advances in Neural Information Processing Systems (NIPS)*, 2009.
- [5] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf, "Measuring Statistical Dependence with Hilbert-Schmidt Norms," in *International conference on Algorithmic Learning Theory*, 2005.
- [6] W. Tang, Z. Lu, and I. Dhillon, "Clustering with Multiple Graphs," in *International Conference on Data Mining*, 2009.
- [7] A. Kumar, P. Rai, and H. Daumé III, "Co-regularized Multi-view Spectral Clustering," *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- [8] E. Schaeffer, "Survey: Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, Aug 2007.
- [9] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, Feb 2010.
- [10] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, Sep 2006.
- [11] —, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun 2006.
- [12] M. Saelens, F. Fouss, L. Yen, and P. Dupont, "The Principal Components Analysis of a Graph, and its Relationships to Spectral Clustering," *European Conference on Machine Learning (ECML)*, 2004.
- [13] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519–524, Mar 1987.
- [14] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, Jan 1990.
- [15] M. Turk and A. P. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Winter 1991.
- [16] A. Edelman, T. A. Arias, and S. T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.
- [17] Y. Chikuse, "Statistics on Special Manifolds," *Lecture Notes in Statistics*, Springer, New York, vol. 174, 2003.
- [18] J. Hamm and D. Lee, "Grassmann Discriminant Analysis: a Unifying View on Subspace-Based Learning," *International Conference on Machine Learning (ICML)*, 2008.
- [19] X. Liu, A. Srivastava, and K. Gallivan, "Optimal Linear Representations of Images for Object Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 662–666, May 2004.
- [20] D. Lin, S. Yan, and X. Tang, "Pursuing Informative Projection on Grassmann Manifold," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [21] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical analysis on Stiefel and Grassmann Manifolds with applications in Computer Vision," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [22] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph Embedding Discriminant Analysis on Grassmannian manifolds for Improved Image Set Matching," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [23] X. Wang, D. Tao, and Z. Li, "Subspaces Indexing Model on Grassmann Manifold for Image Search," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2627–2635, Sep 2011.
- [24] A. Argyriou, M. Herbster, and M. Pontil, "Combining Graph Laplacians for Semi-Supervised Learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.
- [25] D. Zhou and C. Burges, "Spectral Clustering and Transductive Learning with Multiple Views," in *International Conference on Machine Learning (ICML)*, 2007.
- [26] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, Jul 2012.
- [27] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative Matrix Factorization in Multimodality Data for Segmentation and Label Prediction," in *Computer Vision Winter Workshop*, 2011.
- [28] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous Image Feature Integration via Multi-Modal Spectral Clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [29] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "A Regularization Framework for Mobile Social Network Analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.
- [30] D. Eynard, K. Glashoff, M. M. Bronstein, and A. M. Bronstein, "Multimodal diffusion geometry by joint diagonalization of Laplacians," *arXiv:1209.2295*, 2012.
- [31] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-View Clustering via Joint Nonnegative Matrix Factorization," in *SIAM International Conference on Data Mining*, 2013.
- [32] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview Spectral Embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 6, pp. 1438–1446, Dec 2010.
- [33] M. B. Blaschko and C. H. Lampert, "Correlational Spectral Clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [34] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-View Clustering via Canonical Correlation Analysis," in *International Conference on Machine Learning (ICML)*, 2009.
- [35] S. Bickel and T. Scheffer, "Multi-View Clustering," in *IEEE International Conference on Data Mining (ICDM)*.
- [36] A. Kumar and H. Daumé III, "A Co-training Approach for Multi-view Spectral Clustering," *International Conference on Machine Learning (ICML)*, 2011.
- [37] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," in *The 18th Annual Conference on Computational Learning Theory*, 1998.
- [38] V. Sindhwani and P. Niyogi, "A Co-Regularization Approach to Semi-supervised Learning with Multiple Views," in *ICML Workshop on Learning with Multiple Views*, 2005.
- [39] P. Muthukrishnan, D. Radev, and Q. Mei, "Edge Weight Regularization Over Multiple Graphs For Similarity Learning," in *IEEE International Conference on Data Mining (ICDM)*, 2010.
- [40] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with Multi-Layer Graphs: A Spectral Perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, Nov 2012.
- [41] V. R. de Sa, "Spectral Clustering with Two Views," in *ICML Workshop on Learning with Multiple Views*, 2005.
- [42] A. Strehl and J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, Dec 2002.
- [43] E. Bruno and S. Marchand-Maillet, "Multiview Clustering: A Late Fusion Approach Using Latent Models," in *ACM SIGIR Conference on Research and Development on Information Retrieval*, 2009.
- [44] D. Greene and P. Cunningham, "A Matrix Factorization Approach for Integrating Multiple Data Views," in *European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.
- [45] Y. Cheng and R. Zhao, "Multiview Spectral Clustering via Ensemble," in *IEEE International Conference on Granular Computing*, 2009.
- [46] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar 1951.
- [47] F. R. K. Chung, "Spectral Graph Theory," *CBMS Regional Conference Series in Mathematics*, American Mathematical Society, 1997.
- [48] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [49] L. Lovász, "Random Walks on Graphs: A Survey," *Combinatorics, Paul Erdős is Eighty*, vol. 2, pp. 353–398, János Bolyai Mathematical Society, Budapest, 1996.
- [50] G. W. Stewart and J. Sun, "Matrix Perturbation Theory," *Academic Press*, New York, 1990.
- [51] R. Bhatia, "Matrix Analysis," *Springer*, New York, 1997.
- [52] G. H. Golub and C. F. V. Loan, "Matrix Computations (3rd Edition)," *Johns Hopkins University Press*, 1996.
- [53] D. C. Sorensen, "Implicit Application of Polynomial Filters in a k-Step Arnoldi Method," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 357–385, Jan 1992.
- [54] R. B. Lehoucq and D. C. Sorensen, "Deflation Techniques for an Implicitly Restarted Arnoldi Iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, Oct 1996.
- [55] Z. Ghahramani and G. E. Hinton, "The EM Algorithm for Mixtures of Factor Analyzers," *Technical Report CRG-TR-96-1*, University of Toronto, May 1996.
- [56] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," *Cambridge University Press*, 2008.
- [57] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, "Towards rich mobile phone datasets: Lausanne data collection campaign," in *International Conference on Pervasive Services*, 2010.